Clipper

เภา สิงห์บุบผา

SOFT MAIL

สนุกกับ กราฟิก ในคลิปเปอร์

ให้ท่านที่ชอบเขียนโปรแกรมด้วยคลิปเปอร์ ได้รู้เทคนิคการเขียนคำสั่งในด้านกราฟิก

ผู้ที่เขียนโปรแกรมโดยใช้คลิปเปอร์มักเกิดความอึดอัดเมื่อต้องการใช้งานคลิปเปอร์ในโหมดกราฟิก เหมือนกับการโปรแกรมโดยภาษาอื่น ซึ่งคลิปเปอร์ทำไม่ได้ นอกจากจะต้องเชื่อมโยงกับภาษาอื่นซึ่งเป็นที่ทราบว่าทำได้ลำบาก เช่น การเชื่อมโยงกับภาษาซี ก็ต้องไปศึกษาการเขียนภาษาซีด้วย ทำให้เกิดการท้อถอย ผู้เขียนได้พยายามหาเครื่องมือช่วยให้คลิปเปอร์มีความสามารถในการแสดงผลทางด้านกราฟิก พร้อมทั้งประยุกต์การทำงานให้สามารถแสดงผลและรับข้อมูลภาษาไทยกราฟิก 3 ระดับ อีกทั้งการแนะแนวทางในการนำความรู้นี้ไปใช้ในการทำสื่อการเรียนการสอน เพื่อเป็นประโยชน์ต่อวงการศึกษาอีกด้วย หวังว่าคงช่วยให้ผู้ที่ชอบการเขียนโปรแกรมด้วยภาษาคลิปเปอร์ได้เรียนรู้เทคนิคการเขียนคำสั่งในด้านกราฟิก และมีแนวคิดที่กว้างไกลขึ้น อย่างน้อยก็มีภาษาไทยกราฟิกสำหรับคลิปเปอร์ใช้ อันจะทำให้การสื่อความหมายไปยังคนไทยด้วยกันเป็นไปได้ง่ายขึ้น

ผู้เขียนขออธิบายเป็นแนวทางดังนี้

1. การนำ DGE มาใช้งานร่วมกับคลิปเปอร์ 5.01
2. การแสดงผลกราฟิกต่าง ๆ
3. การนำภาพฟอร์แมต PCX และ ICON มาแสดงผล
4. การแสดงผลภาษาไทยกราฟิก 3 ระดับ
5. การรับข้อมูลภาษาไทยของคลิปเปอร์ในกราฟิกโหมด

ซึ่งหากมีโอกาส ผู้เขียนจะกล่าวถึงการนำคลาสไลบรารีของ Visual interface For CLIPPER และ Class(Y) ซึ่งจะเป็นการเขียนโปรแกรมในแบบ OOP ผู้เขียนโปรแกรมโดยคลิปเปอร์จะได้มีโอกาสสัมผัสกับการเขียนโปรแกรมแบบ OOP ที่ว่าเขียนง่ายนั้นเป็นอย่างไร

การปรับแต่ง DGE เพื่อใช้กับคลิปเปอร์ 5.01

ผู้อ่านอาจคิดว่า ในขณะนี้โปรแกรมคลิปเปอร์ได้ถูกปรับปรุงไปจนถึงเวอร์ชัน 5.2 แล้ว เหตุใดผู้เขียนจึงยังคงใช้รุ่น 5.01 ที่เป็นเช่นนี้เพราะว่าในรุ่น 5.2 ของบริษัท Computer Associate ที่ปรับปรุงจากคลิปเปอร์รุ่น 5.01 นั้น เท่าที่ผู้เขียนนำรุ่น 5.2 มาใช้งานจะใช้ร่วมกับไลบรารีบางตัวไม่ได้ ผู้เขียนจึงยังคงใช้รุ่น 5.01 อยู่

1. ผู้อ่านสามารถทบทวนเกี่ยวกับ DGE ได้จากไมโครคอมพิวเตอร์เล่ม 87 ซึ่งในการติดตั้ง DGE ในรุ่นที่ 4.0 นั้นใช้การ INSTALL แทน ทำให้ง่ายในการติดตั้ง เมื่อผู้อ่านติดตั้งโปรแกรมแล้ว จะต้องระเบิดแฟ้ม 2 แฟ้มที่ได้บีบอัดไว้คือ DGENGARC.COM จะได้แฟ้ม DGE.NG ซึ่งเป็น HELP ที่ใช้กับนอร์ตันไกด์ ให้นำแฟ้มนี้ไปเก็บไว้ในสับไดเรคทอรี \NG เพื่อไว้เรียกดู อีกแฟ้มที่บีบอัดไว้คือ FONTARC.EXE เมื่อขยายจะได้แฟ้ม GFONT.EXE และฟอนต์ ซึ่งสามารถนำมาแก้ไขรูปแบบฟอนต์และไอคอนได้ ในการติดตั้งนี้จะได้ไอคอนตัวอย่าง คือ HELP.ICO ด้วย

2. ให้ติดตั้ง DGE ในสับไดเรคทอรี C:\DGE แล้วสร้างสับไดเรคทอรี C:\DGE\LIB นำแฟ้ม DGE.LIB มาไว้ในสับไดเรคทอรีนี้

3. สร้างสับไดเรคทอรี C:\DGE\OBJ นำแฟ้ม DGEVGA.OBJ และ NULEXVGA.OBJ มาเก็บไว้

4. กำหนดสภาวะแวดล้อมให้คลิปเปอร์ดังนี้ (แก้ไขเพิ่มเติมใน AUTOEXEC.BAT)

```
PATH=C:\CLIPPER\BIN;.....
SET LIB=C:\CLIPPER\LIB;C:\DGE\LIB
SET OBJ=C:\CLIPPER\OBJ;C:\DGE\OBJ
SET INCLUDE=C:\CLIPPER\INCLUDE
SET PLL=C:\CLIPPER\PLL
```

เมื่อกำหนดสภาพแวดล้อมแล้ว เวลาคอมพายล์โดย CLIPPER และลิงค์โดยใช้ RTLINK จะหาแฟ้ม INCLUDE,LIB และ OBJ ตามพาธที่กำหนด ทำให้ง่ายต่อการใช้งาน

สำหรับผู้ที่ไม่สามารถหาโปรแกรม DGE มาใช้งานได้ ก็สามารถใช้บริการทางซอฟต์แมล์ โดยผู้เขียนได้จัดเตรียมไฟล์ OBJ ให้ 3 ตัวคือ DGEVGA.OBJ, NULEXVGA.OBJ และ DGEKERN.OBJ สามารถนำไปลิงค์แทนไลบรารี DGE.LIB อีกทั้งมีโปรแกรมตัวอย่างให้ด้วย

ลักษณะกราฟิกบนจอภาพ

การเข้าสู่สภาวะกราฟิกของ DGE ใช้ฟังก์ชัน SETHIRES(0) เมื่อเรียกใช้แล้วจะเข้าสู่สภาวะกราฟิก การกำหนดตำแหน่งจะเปลี่ยนไป โดยมุมซ้ายล่างเป็น 0,0 และขวาบนเป็น 1350, 1000 ดังแสดงในรูปที่ 1 จะใช้คำสั่ง ROW, COLUMN ในสภาวะเท็กซ์โหมดไม่ได้

การเปลี่ยนจากกราฟิกโหมดเป็นเท็กซ์โหมด ใช้ฟังก์ชัน SETTEXT( )

การสร้างวงกลม

เมื่อต้องการสร้างภาพวงกลมนั้นสามารถทำได้โดยการเข้าสู่กราฟิกโหมดแล้วใช้ฟังก์ชัน

```
drawcircle (X, Y, Rad, Ang1, Ang2, Mode, Style, Color)
```

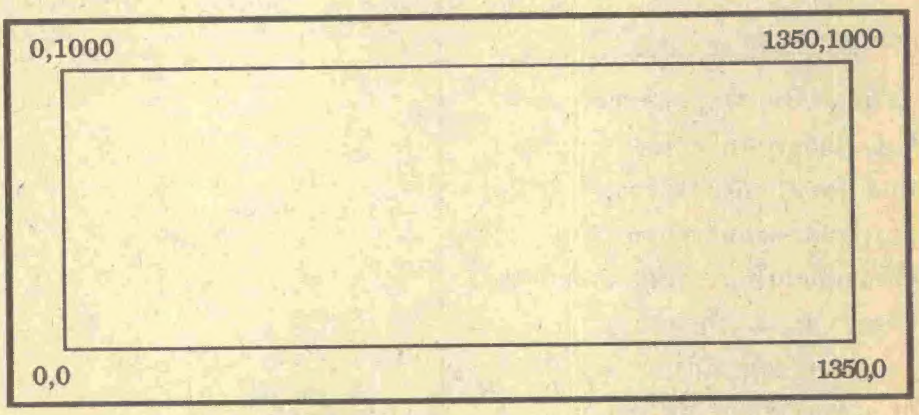
X และ Y คือ จุดศูนย์กลางของวงกลม

Rad คือ รัศมีของวงกลม

Ang1 คือ มุมที่เริ่มสร้างวงกลม

Ang2 คือ มุมสิ้นสุดของวงกลม

Mode คือ กำหนดสภาพการวาด



รูปที่ 1

Style คือ เป็นการกำหนดลวดลายสีพื้นของวงกลม มีค่า 0-15

Color คือ การกำหนดสีมีค่า 0-15

เราสามารถสร้างวงกลม วงกลมมีลวดลาย หรือส่วนของวงกลมก็ได้ ขอให้ศึกษาการทำงานของโปรแกรมจาก CIRCLE.PRG ซึ่งอยู่ในรูปที่ 2 เป็นตัวอย่าง

ผลจากการรันโปรแกรมในรูปที่ 2 จะได้ดังรูปที่ 3

```
//
// Program circle.prg
// Compile  : CLIPPER circle
// LINK     : RTLINK FILE circle,dgevga,nulexvga,dgekern
//          : RTLINK FILE circle,dgevga,nulexvga LIB dge
// RUN      : CIRCLE
//
clear
sethires(0)
pattern = 15
drawcircle(300, 500, 200,  0, 360, 8, pattern, 0)
drawcircle(700, 500, 200, 60, 120, 8, pattern, 0)
WAIT ""
SETTEXT()
RETURN
```

รูปที่ 2 โปรแกรม CIRCLE.PRG

การสร้างสี่เหลี่ยม

การสร้างสี่เหลี่ยมเมื่อเข้าสู่กราฟิกโหมด จะใช้ฟังก์ชัน BOXFILL( ) (รูปที่ 4) มีรูปแบบดังนี้

```
boxfill (X, Y, Wid, Ht, Patt, Color)
```

X และ Y เป็นจุดล่างซ้ายของสี่เหลี่ยม

Wid คือ ความกว้างของรูปสี่เหลี่ยม

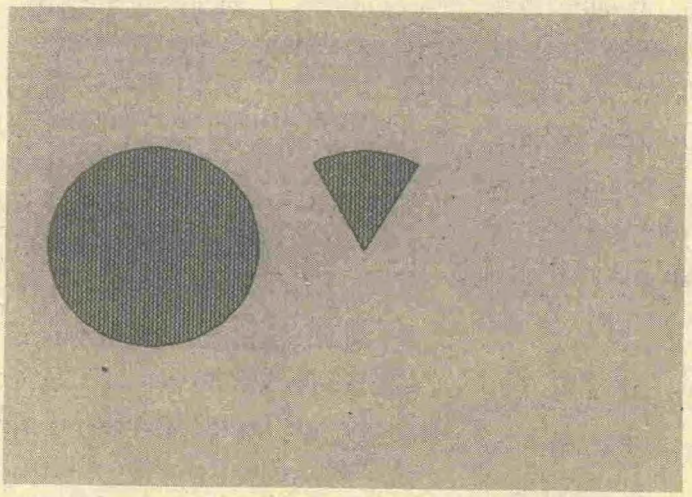
Ht คือ ความสูงของรูปสี่เหลี่ยม

Patt คือ ลวดลายของพื้นสี่เหลี่ยม มีค่า 0-15

Color คือ การกำหนดสีของสี่เหลี่ยม มีค่า 0-15

สำหรับตัวอย่างการสร้างรูปสี่เหลี่ยมต่อไปนี้จะเป็นการตรวจสอบแพตเทอร์นด้วย โดยจะแสดงสี่เหลี่ยมซ้อนกันทั้งหมด 16 รูป (ดังแสดงในรูปที่ 5) แต่ละรูปจะแสดงแพตเทอร์นต่าง ๆ จะเป็นประโยชน์ในการกำหนดแพตเทอร์นของการแสดงกราฟิกแบบอื่น ๆ

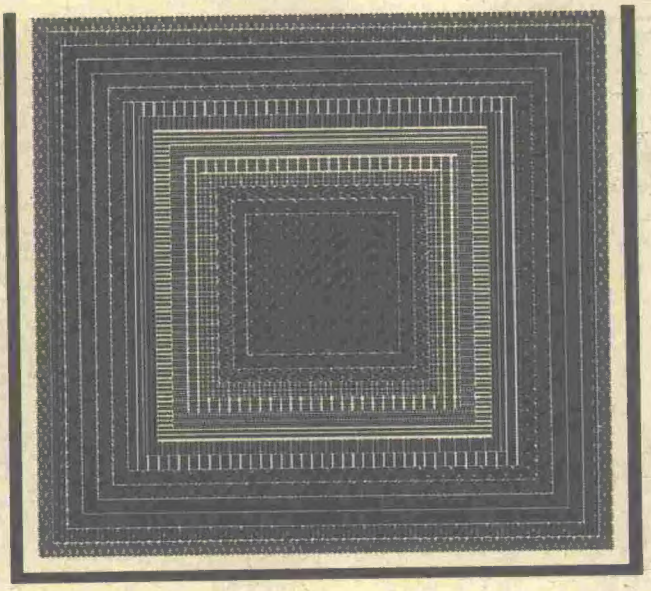
จากตัวอย่างโปรแกรมในรูปที่ 4 ผู้อ่านสามารถดัดแปลงเป็นกรอบ หรือ BUTTON ได้



รูปที่ 3 เป็นผลจากการรันโปรแกรม CIRCLE. PRG

```
//
// Program box.prg
// Compile : CLIPPER box
// LINK    : RTLINK FILE box,dgevga,nulexvga,dgekern
//         : RTLINK FILE box,dgevga,nulexvga LIB dge
// RUN     : BOX
//
clear
sethires(0)
pattern = 0
x = 20
y = 1000
DO WHILE pattern <= 15
  boxfill(x, x, y, y, pattern, 14)
  x = x+25
  y = y-50
  pattern++
ENDDO
WAIT ""
SETTEXT()
RETURN
```

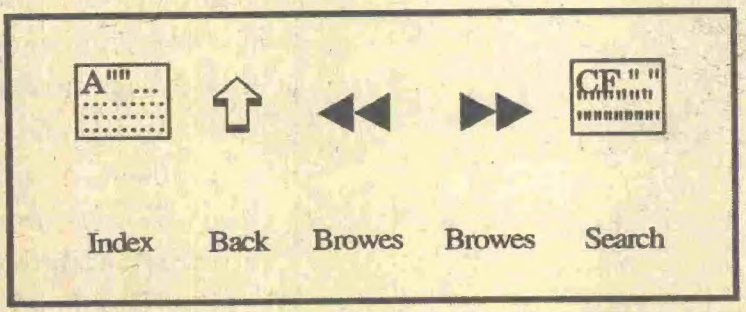
รูปที่ 4 โปรแกรม BOX.PRG



รูปที่ 5 ตัวอย่างการสร้างรูปสี่เหลี่ยมที่ใช้ในการตรวจสอบแพดเทอร์น

```
//
// PROGRAM icon.prg
// COMPILE : CLIPPER icon
// LINK    : RTLINK FILE icon,dgevga,nulexvga LIB dge
//         : RTLINK FILE icon,dgevga,nulexvga,dgekern
// RUN     : ICON
//

sethires(0)
LOADICON("help.ico")
x = 100
y = 500
icon = 16
mode = 4
DO WHILE icon <= 20
   drawicon(x, y, mode, icon, 7)
//   drawicon(500, 500, 4, 17, 7)
   icon = icon + 1
   x = x + 100
ENDDO
WAIT ""
SETTEXT()
RETURN
```

รูปที่ 6 โปรแกรม ICON.PRG



รูปที่ 7 แสดงไอคอน 5 ตัว

การนำไอคอนมาแสดง

ในตัวอย่างของโปรแกรมจะมีแฟ้ม HELP.ICO (รูปที่ 6) ซึ่งเป็นไลบรารี-ไอคอน ซึ่งสามารถมาแก้ไขหรือสร้างใหม่โดยยูทิลิตี้ GFONT.EXE ซึ่งมีมาพร้อมกับ DGE แต่ไม่มีในซอฟต์เมล์ ใน ICON แต่ละแฟ้มจะเก็บไอคอนไว้ 8 ตัวด้วยกัน เริ่มตั้งแต่ตัวที่ 0-7 ไอคอนสามารถนำมาแสดงเพื่อเป็นการสื่อความหมาย ขั้นแรกต้องมีการโหลดแฟ้ม ICON ก่อนด้วยฟังก์ชัน LOADICON( )

```
loadicon (File)
```

file คือ ชื่อแฟ้มไอคอน ถ้าไม่อยู่ในสับไดเรคทอรีเดียวกันต้องบอกพาธด้วย เช่น

```
loadicon ("HELP.ICO")
loadicon ("C: \ YOURSUB \ HELP.ICO")
```

เมื่อโหลดแฟ้มไอคอนแล้ว จะแสดงภาพไอคอนโดยใช้ฟังก์ชัน DRAW-ICON( )

```
drawicon (X, Y, Mode, IconNo, Color)
```

X, Y คือ ตำแหน่งแสดงของไอคอน

Mode คือ การกำหนดการแสดง เช่น การซ้อนทับ การลบภาพเดิมแล้วแสดง มีค่าตั้งแต่ 0-127 ปกติใช้ 4

IconNo คือ ลำดับที่ของไอคอนในแฟ้มไอคอนที่นำมาแสดง ให้บวกด้วย 16

Color คือ สีของไอคอน

ในตัวอย่างแฟ้ม HELP.ICO จะมีไอคอนมาให้ 5 ตัว ดังแสดงในรูปที่ 7 การเรียกใช้งานจึงเรียกตั้งแต่ 15+0 จนถึง 15+5 คือ 15 ถึง 20 นั่นเอง

```
//
// PROGRAM pcx.prg
// COMPILE  : CLIPPER pcx
// LINK     : RTLINK FILE pcx,dgevga,nulexvga LIB dge
//          : RTLINK FILE pcx,dgevga,nulexvga,dgekern
// RUN      : pcx
//

CLEAR
SETHIRES(0)
PICREAD(100,100,0,"BRAIN.PCX")
WAIT ""
SETTEXT()
RETURN
```

รูปที่ 8 โปรแกรม PCX.PRG

รูปที่ 9

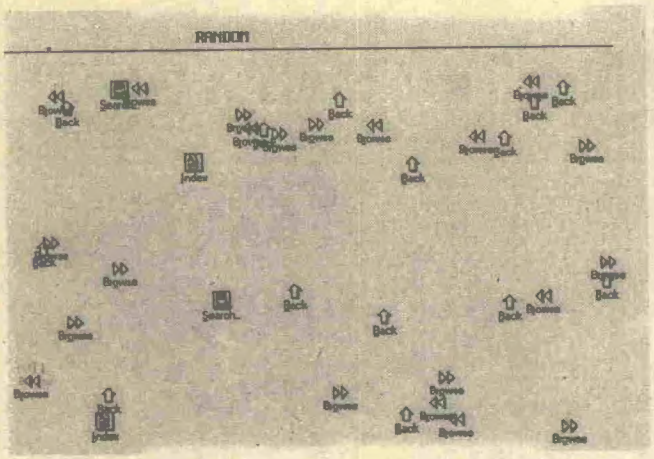
```
//
// program b_random.prg
// COMPILE  : CLIPPER b_random
// LINK     : RTLINK FILE b_random,nulexvga,dgevga,
//                        random LIB dge OUT random
//          : RTLINK FILE b_random,nulexvga,dgevga,
//                        random,dgekern OUT random
// RUN      : RANDOM

#include "inkey.ch"
clrscreen()
sethires(0)
saystring(400,920,8,0,12,"RANDOM")
drawline(0,910,1280,910,0,0,2)
LOADICON("help.ico")
DO WHILE !key_pressed(K_ESC)
   x     = ROUND(RANDOM( 10,1280, 0),0)
   y     = ROUND(RANDOM( 10, 850, 0),0)
   color = ROUND(RANDOM(  0,  15, 0),0)
   icon  = ROUND(RANDOM( 16,  20, 0),0)
   DRAWICON(x,y,4,icon,color)
ENDDO

CLRSCREEN()

DO WHILE !key_pressed(K_ESC)
   x     = ROUND(RANDOM(10,1280,0),0)
   y     = ROUND(RANDOM(10, 960,0),0)
   rad   = ROUND(RANDOM(0,200,0),0)
   ang1  = ROUND(RANDOM(0,360,0),0)
   ang2  = ROUND(RANDOM(0,360,0),0)
   mode  = ROUND(RANDOM(8,15,0),0)
   style = ROUND(RANDOM(0,1,0),0)
   color = ROUND(RANDOM(0,15,0),0)
   DRAWCIRCLE(x,y,rad,ang1,ang2,8,1,color)
ENDDO
SETTEXT()
RETURN
FUNCTION key_pressed(key)
RETURN inkey() = key
```

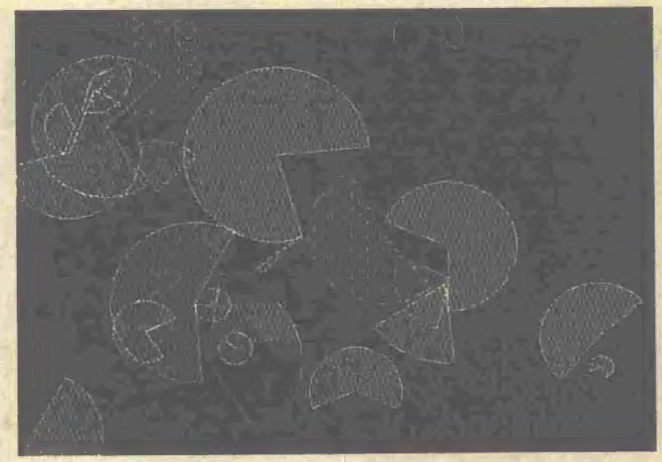
รูปที่ 10
โปรแกรม RAMDOM.PRG

การแสดงภาพ PCX

ภาพในฟอร์แมต PCX (รูปที่ 8) มาจากโปรแกรมเพนต์บรัช ซึ่งอาจจะได้จากการวาดขึ้นมาหรือมีอยู่แล้ว

จะเห็นได้ว่าการนำภาพ PCX มาแสดงนั้น เมื่อใช้ฟังก์ชันใน DGE มาช่วย จะมีความง่ายมาก สำหรับรูปที่ 9 นั้น สังเกตว่าจะมีคำอธิบายเป็นภาษาไทยอยู่ด้วย เป็นการแสดงผลภาษาไทยกราฟิกร่วมกับการแสดงภาพ ผู้อ่านอาจจะมองแนวทางในการประยุกต์นำไปใช้งานอย่างอื่นอีก เช่น การแสดงประวัติบุคคล การทำสื่อการเรียนการสอนในวิชาต่าง ๆ สำหรับการแสดงผลภาษาไทยในกราฟิกโหมด ผู้เขียนขอยกยอดไว้ในฉบับต่อไป





รูปที่ 11

ของแถมฟังก์ชัน RANDOM( ) สำหรับคลิปเปอร์

ในบางครั้งผู้เขียนโปรแกรมโดยคลิปเปอร์ต้องการค่าที่สุ่มมาเพื่อใช้งาน แต่ฟังก์ชันนี้ไม่มีในคลิปเปอร์ ให้เรียกใช้เหมือนกับภาษาอื่น แม้แต่ในเบสิกยังมี ผู้เขียนจึงขอแถมฟังก์ชัน RANDOM( ) ดังแสดงในรูปที่ 10 ซึ่งเขียนจากภาษาซีเพื่อนำมาลิงค์กับคลิปเปอร์ (เรียกใช้ได้ทั้งรุ่น SUMMER และ 5.01) ผู้เขียนจัดให้ในซอฟเมล์ และจะมีตัวอย่างโปรแกรม B_RANDOM.PRG เป็นการสุ่มตัวไอคอน สี ตำแหน่ง และการสุ่มการวาดส่วนของวงกลม ให้ทดลองคอมพายล์ ลิงค์ และรันโปรแกรมดู

```
RANDOM (expN1, expN2, expN3)
```

เช่น number = RANDOM (0, 10, 0) จะได้ค่าตัวแปร number ที่สุ่มระหว่าง 0 ถึง 10 ซึ่งจะให้ค่าตัวเลขระหว่าง expN1 และ expN2 พร้อมทั้งทศนิยมอีก 2 ตำแหน่ง หากผู้อ่านต้องการเลขจำนวนเต็มให้ใช้ฟังก์ชัน ROUND( ) ช่วย

เมื่อทดลองรันโปรแกรมจะเป็นการสุ่มตัวไอคอน เมื่อกด Esc จะเข้าสู่การสุ่มการวาดส่วนของวงกลมดังรูปที่ 11

ส่งท้าย

สำหรับคำอธิบายการทำงานของแต่ละฟังก์ชัน อาจไม่ชัดเจนเพราะจะทำให้เนื้อหายาวมาก ขอให้ผู้เขียนลองศึกษาฟังก์ชันของ DGE (มีประมาณ 100 ฟังก์ชัน) จากนอร์ตันไกด์ ซึ่งยังมีฟังก์ชันอื่น ๆ ที่มีประโยชน์ในการแสดงผลทางด้านกราฟิกอีกมาก เช่น การวาดเส้น การแสดงกราฟแบบต่าง ๆ การติดต่อกับเม้าส์ เป็นต้น ในตอนหน้าผู้เขียนขอเสนอหลักการแสดงผลภาษาไทยกราฟิกในคลิปเปอร์